

Utah State University

DigitalCommons@USU

---

Undergraduate Honors Capstone Projects

Honors Program

---

5-2003

## Final Design Report of Computer Controller of Robotic Arm

Kurt Niederhauser  
*Utah State University*

Follow this and additional works at: <https://digitalcommons.usu.edu/honors>



Part of the [Robotics Commons](#)

---

### Recommended Citation

Niederhauser, Kurt, "Final Design Report of Computer Controller of Robotic Arm" (2003). *Undergraduate Honors Capstone Projects*. 855.

<https://digitalcommons.usu.edu/honors/855>

This Thesis is brought to you for free and open access by the Honors Program at DigitalCommons@USU. It has been accepted for inclusion in Undergraduate Honors Capstone Projects by an authorized administrator of DigitalCommons@USU. For more information, please contact [digitalcommons@usu.edu](mailto:digitalcommons@usu.edu).



FINAL DESIGN REPORT  
OF  
COMPUTER CONTROLLER OF ROBOTIC ARM

by

Kurt Niederhauser

Thesis (Report) submitted in partial fulfillment  
of the requirements for the degree

of

DEPARTMENT HONORS

(OR)

UNIVERSITY HONORS  
WITH DEPARTMENT HONORS

in

Computer Engineering

Approved:

---

Thesis/Project Advisor

---

Department Honors Advisor

---

Director of Honors Program

UTAH STATE UNIVERSITY  
Logan, UT

2003

Final Design Report

Of

Computer Controller  
Of Robotic Arm

(Application With T4 Mobile Robot)

My Part of the project was to design the hardware. I assembled the TT8 microprocessor and designed the optocoupler isolation board for the input and output signals to the TT8 microprocessor. We had two boards. One board had the TT8 Processor and other chips used for regulating the affairs of the TT8 (This board I assembled), and then there was another board that had all the control signals for the TT8 (this board I had to design), which sent the signals letting the TT8 know when the robot arm had been extended or retracted as far as it could go. Dave, my colleague, did all the software for this project, programming the TT8 to use my input signals and halt the extend and retract clock pulse governing the motors to the robot arm.

## **SUMMARY:**

The project we undertook was the design and building of the *Computer Controller of the Robotic Arm* for use with the T4 Mobile Robot, currently being designed by the USU Center for Self-Organizing and Intelligent Systems (CSOIS). We proposed to design and build the all circuits necessary to identify the arm's position and correctly drive the robotic arm motors. This included designing circuits that both provide the power necessary to run the motors, and also protected the rest of the system against power surges. We developed the software necessary to control the robotic arm through data packets received from the command source (joystick or T4 central processor).

### **Project Design Team**

David Nickle

Kurt Niederhauser

## **CONTENTS:**

1.0	Introduction and General Project Summary	4
2.0	Review of Preliminary Design	6
3.0	System Design	11
3.1	Hardware Schematics	12
3.2	Software Design Code	18

## **Introduction and General Project Summary:**

The USU Center for Self-Organizing and Intelligent Systems (CSOIS) is currently involved in the development of the *T4 Mobile Robot*. The T4 robot is being designed as a nearly autonomous mobile security system. It is projected to be used for parking lot security to scan and identify authorized and unauthorized vehicles. The project already includes the development of a mini-rover to check the undercarriage of vehicles. The *Robotic Arm* extension of the project we helped develop was to allow for further examination of suspicious vehicles by a camera mounted at the end of the arm.

In order for the T4 Mobile Robot project to contain all the tools necessary for a thorough examination of a suspicious vehicle, it needs the capability to examine the vehicle from all side and different angles. The mini-rover will provide the ability to view the vehicle from underneath. Our project – the Robot Arm – should provide the utility to see the vehicle from above and be able to peer into the interior of the vehicle. It should be maneuverable and responsive.

A Robotic Arm has been developed by a group of Mechanical Engineers for use with this project. It allows for vertically raising and lowering the angle of the arm and will extend and collapse the length of the arm. This design is driven by three motors. The position of these motors is indicated by analog potentiometers. There are limit switches installed to indicate full extension and full collapse. This mechanical design lacked the computer-controlling design to integrate the components into a working whole. By our participation in the project, we developed a sound computer design to work with what has been developed by the Mechanical Engineers.

Our design was centered around a TT8 processor. The processor receives command packets from the outside control module (*Joystick or T4 Main Controller*) through a RS232 serial port, effectively carries out those commands, and sends status packets back to the outside control module, again using the RS232 serial port. The driver circuits we designed protect the processor from possible power surges from the high voltage necessary to drive the motors.

Hardware protection is achieved by optical isolation of the connections between the TT8 processor and all off-board hardware, including: all signals to the IB462 stepper motor drivers and limit switches. The design also contains a hardware implemented shutdown of driver signals, should the main processor fail or freeze. See schematics for design specifics.

The software developed in this project runs under the uCOS real-time operating system. Priority levels and the frequency of function calls allow for time required execution of all tasks. Testing to this point has shown that system resources are adequately protected between the threads to prevent software errors.

Testing of the controller on the robot arm has shown it to be effective in correctly driving the motors and responding to position and sensor input. Our design of the controller however does not solve all the problems inherent in the physical layout and properties of the arm: i.e. the weight of the arm and tendency of the extension/retraction cable to cramp and bind.



## **Review of Preliminary Design:**

- 1) We proposed to design safe and adequate driver circuits for the motors and other components:
  - a. The stepper-motor drivers require several input signals from the processor. The mechanical design of the robotic arm we are working with uses motors driven by a 24 to 40 V source. There is a potential risk of an unfortunate short sending a high amount of current back through the driver and into the processor, which could blow out the processor. We needed to protect against that possibility.
  - b. We needed to design circuits for the potentiometers that will provide and clear and usable readings to the processor.
- 2) Software Design:
  - a. Whatever the controlling source, the TT8 processor controlling the arm will receive packets containing the command data through the RS232 port. The packets are comprised of a header and a body containing data to indicate the desired position. We were to design software to allow the processor to efficiently recognize and deal with these packets.
  - b. Reporting back. The software we designed needed to also report its status information back to the controller so that the controller can utilize that information if desired.
- 3) Limits Switches:
  - a. There are limit switches on how far the robot arm will be allowed to move up and down, and when the arm is fully extended or retracted. This reduces unnecessary stress on the mechanical arm and motors. We were to recognize signals from the

limit switches and design effective procedures for what to do when the processor receives these signals.

4) Controlling the extension and retraction motors together:

- a. There are two motors which control the extend and retract motions of the robot arm. One motor is made to control the extend motion and another motor is to control the retract motion. The two motors have a two to three gear ratio difference between the two pulleys for the cables. The stepper motor controller must send the same clock tick and direction to both motors but allow for both motors to stay in sync.

5) Computer Controller:

- a. At the end of the project we were to have built a controller that interfaces with the outside controlling module and runs the motors and indicators of the robot arm's mechanical design to their potential.

## **Summary of Decision Analysis:**

### **Hardware:**

To solve the problem of hardware protection, we foresaw two alternatives available to us:

One was to use drivers with a high reverse impedance on the outputs and place pulldown resistors on all the signal lines. In this case should a backward current flow occur, we hope that the path through the resistors can sink enough current to protect the hardware components from damage.

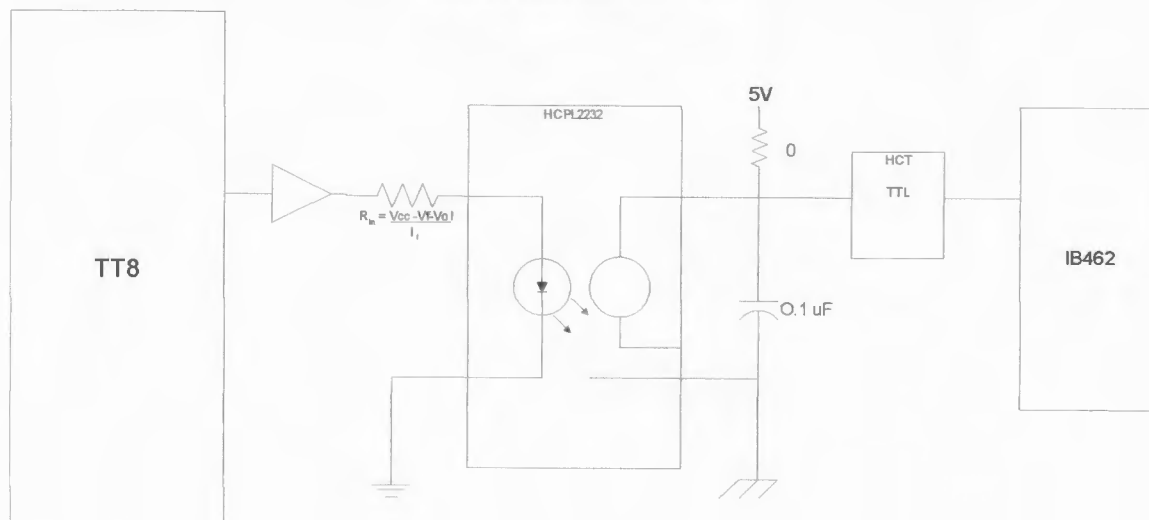
The other was to use optoisolators (or optocouplers). Optocouplers propagate signals by light and sensors. This effectively makes a break in the transmission lines that

a reverse current cannot cross, electrically isolating the hardware on one side of the chip from the hardware on the other.

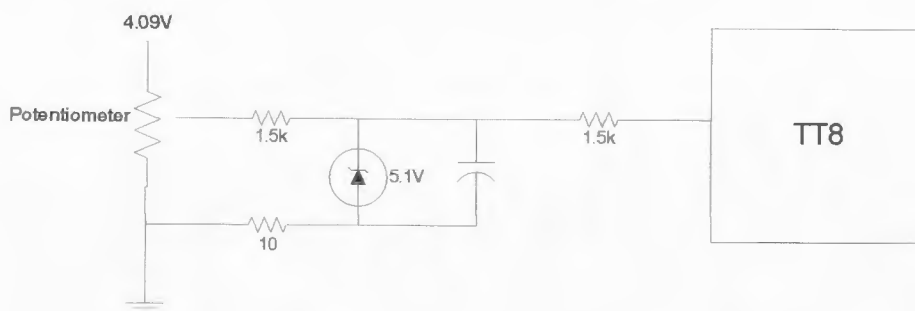
The advantage to the first method was that the components were somewhat cheaper. It would be effective under most circumstances. But we decided on using the optoisolators because for low speed propagation (under 1 megabits/sec) the chips are quite cheap. Our speeds would be well under that range. Plus, it adds a bit more fail safe security.

We designed the following circuits to isolate the motor driving signals and handle the analog signal from the potentiometer. We would place a motor interface circuit on each output.

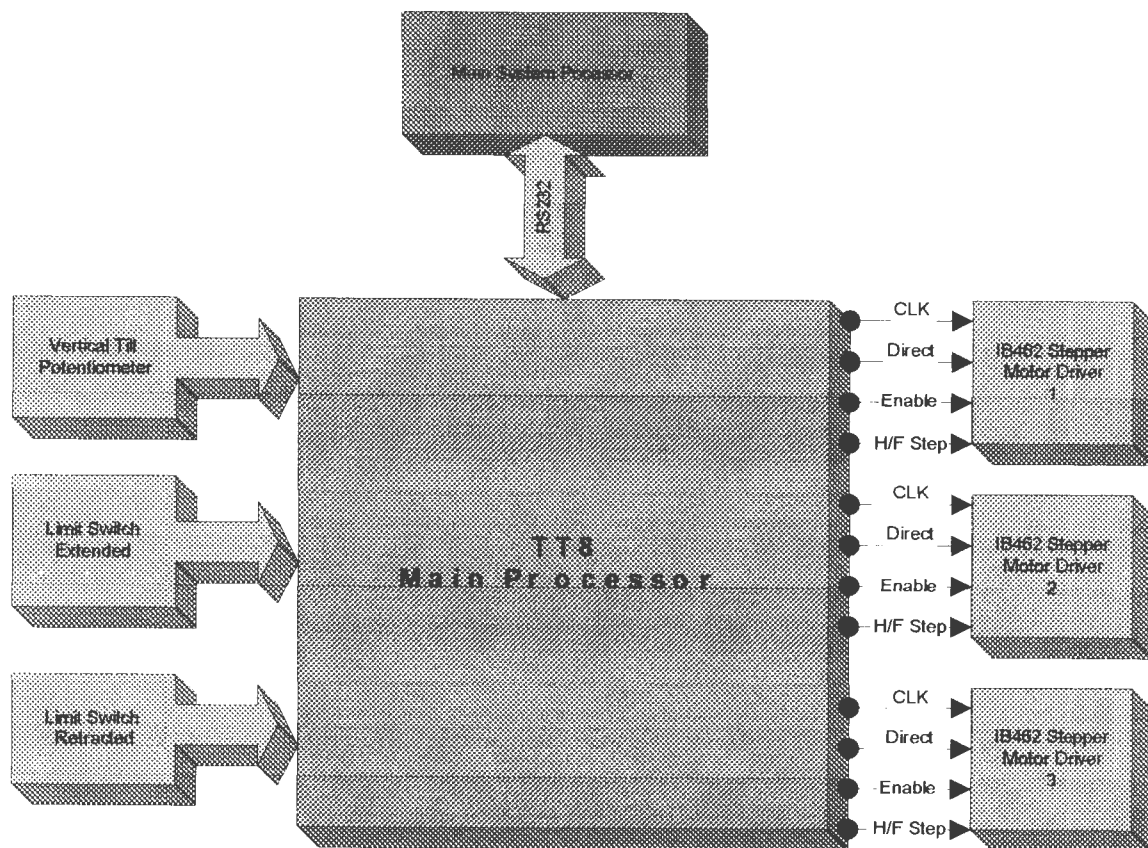
Motor Interface Circuit



Potentiometer Input Circuit



We originally drew up the signal layout like this:



We expected we would have to protect 15 signal lines: 4 outs for each motor and 3 inputs.

When we presented our design to our development supervisor, he had on hand a board that provided for mounting a TT8 and brought out the lines for using the AtoD ports on the processor. But it only gave us 6 output lines. We felt the final product would be much more compact and efficient if we used this board. So we decided, since two of the motors worked together, we could combine their signals into one; using the same clock signal and inverting the direction signal to one of the motors. We also decided to tie the motors to full step function. This would give us greater torque and the accuracy gained through half stepping is not so vital to our use. And then finally we decided that it is best to always leave the motors enabled whenever the processor is on, since disabling them while in use would cause the arm to fall. So we tied those signals. This left us requiring one CLK and one direction signal for two of the motors, and a CLK and direction signal for the other. That gave us two free TPU outputs that we could use for accelerating the stepper motors or for another motor if we wanted. The board he suggested also had two digital inputs that we decided to use for the limit switches.

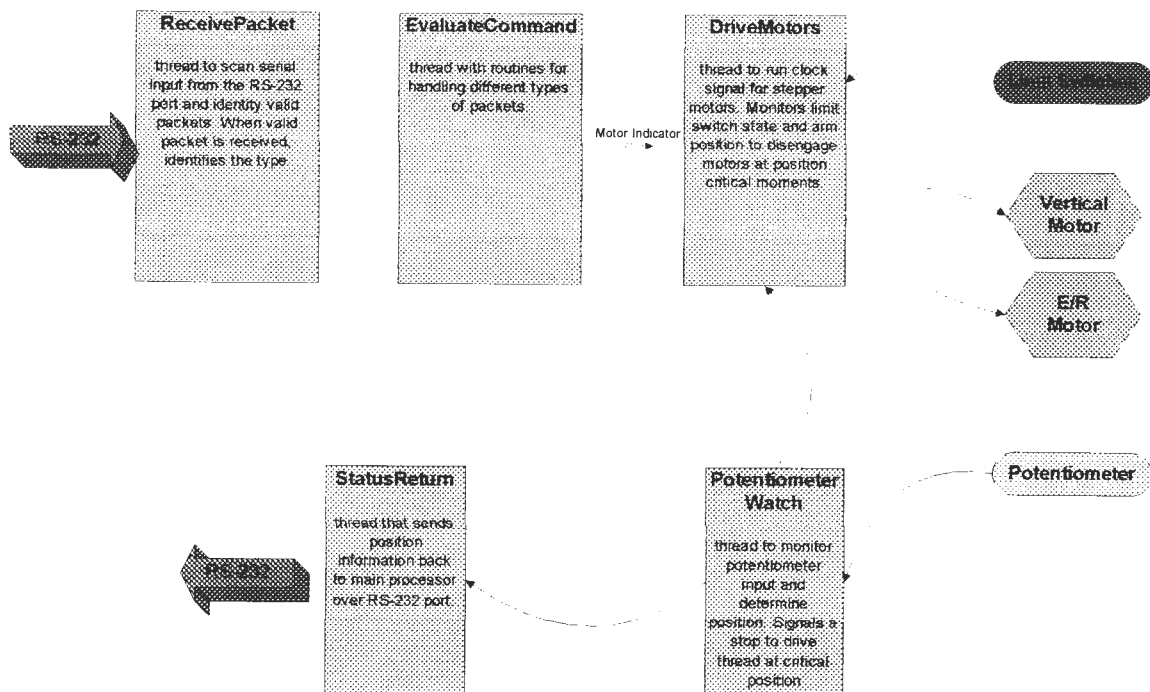
Another thing we considered was how many potentiometers to use. We definitely had to have one on the vertical motor since it would probably only want it to extend between 0 and 60 degrees. We thought about placing another potentiometer on the extend motor identify its position. The decision we arrived at, however, was that it would be less specific and redundant. Firstly, full extension requires several revolutions of the motor. We could include a counting loop and determine position by how many times the motor has made a complete revolution, but if an operator wanted to inspect a vehicle, he would probably be operating the camera manually, so the extension mechanism would not have to operate autonomously. Specific position wouldn't be as important as function. And the limit switches would protect the arm from going beyond its boundaries.

### Software:

The software we needed to develop had to be able to receive and recognize a command packet, decipher the information contained in that packet, control the motors accordingly and send back the status of the arm. We also needed to monitor the position and limit signals.

These functions would need to work together like this:

## THREAD DIAGRAM



## **Design of System:**

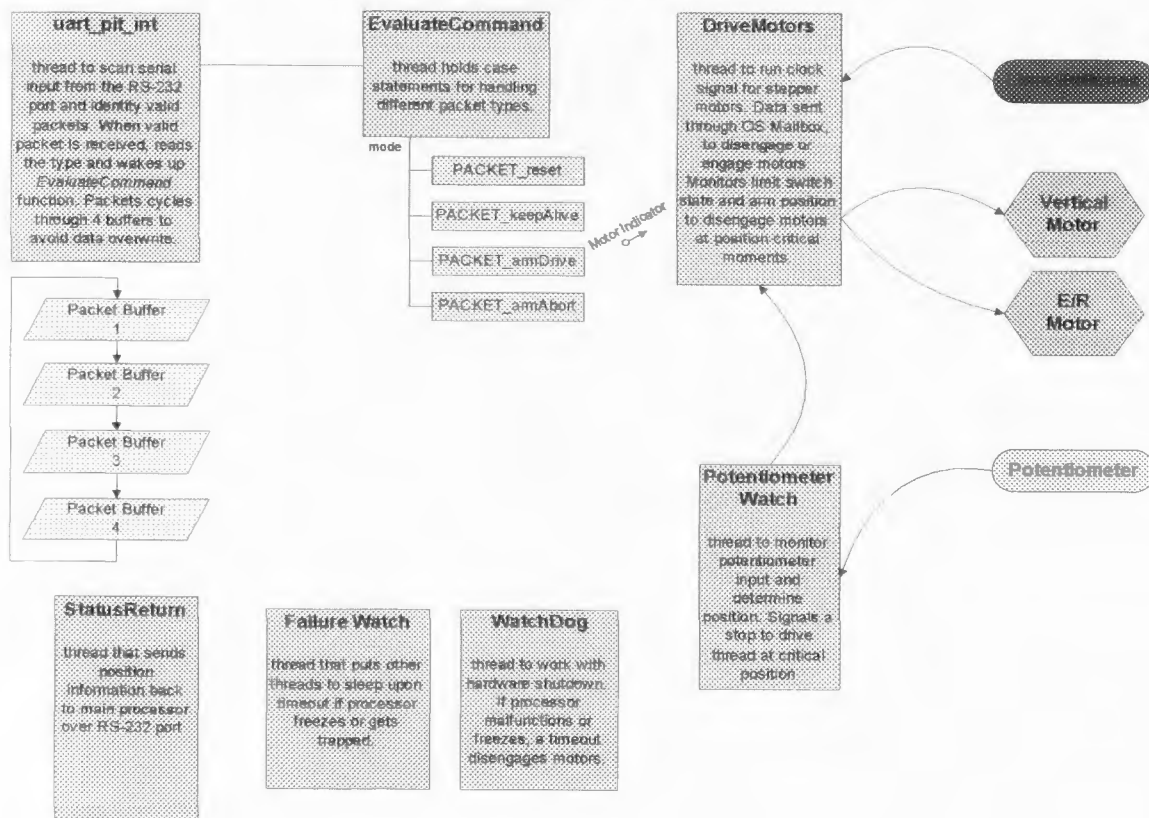
### **Hardware:**

In building our system, we included the chip layout on two separate boards. This allowed us to take advantage of the pre-cast board to hold the TT8, and the other contains the HCPL-2531 optoisolators and peripheral interface connections to the motor drivers, potentiometers and limit switches. The schematics diagrams show the building instructions for two separate boards, but they may combined onto one breadboard if necessary.

## Software:

In the final software design, we decided to add a couple of additional control threads to take advantage of the additional hardware safe guards present on the TT8 board. The current software design functions like this:

### THREAD DIAGRAM



Communication between the threads is handled by messaging using mailboxes and an information structure called RobArmStat (Robot Arm Status) that maintains information about the current state of the arm motors and position.

We also needed to design the packets structure we would expect to receive for controlling the arm, and the packet structure we would send back to indicate the arm status. The packet structures are designed as follows:

## Packet Structure

Arm Drive Packet:



Arm Status Packet:



For indicator packets, such as a motor stop packet, or error packets such as a limit switch signal, the same structure is used. The field values, however, will contain data that indicates a difference in operation.

The software code is provided on the following pages for reference:



We have included an estimated time schedule and rough system diagram in the Appendix.

**Conclusion:**

Upon completion of the project we will have produced a computer controller system that can be used with the existing robot arm design. It will be able to receive and interpret the kind of command packets designated by the CSOIS T4 Mobile Robot Design Team. It will drive the motors to their capacity to manipulate the arm correctly.

If there are further requirements I have not identified in this proposal, I will be happy to discuss them with you and add them to our design process. Thank you.

Kurt Niederhauser

Design Team Partner